Nuclear Engineering and Technology xxx (xxxx) xxx

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

Evaluation of effectiveness of fault-tolerant techniques in a digital instrumentation and control system with a fault injection experiment

Man Cheol Kim [a], Jeongil Seo [b], Wondea Jung [c], Jong Gyun Choi [c], Hyun Gook Kang [d], Seung Jun Lee [b, *]

[a] Chung-Ang University, 84 Heukseok-ro, Dongjak-gu, Seoul, 156-756, South Korea
[b] Ulsan National Institute of Science and Technology, 50 UNIST-gil, Ulju-gun, Ulsan, 44919, South Korea
[c] Korea Atomic Energy Research Institute, 1405 Daedeok-daero, Yuseong-gu, Daejeon, 305-353, South Korea
[d] Rensselaer Polytechnic Institute, 110 8th St, Troy, NY, 12180, USA

ARTICLE INFO

Article history:
Received 28 September 2018
Received in revised form
31 October 2018
Accepted 23 November 2018
Available online xxx

Keywords:
Digital I&C system
Probabilistic safety assessment
Fault injection
Fault-tolerant technique
Fault detection coverage

ABSTRACT

Recently, instrumentation and control (I&C) systems in nuclear power plants have undergone digitalization. Owing to the unique characteristics of digital I&C systems, the reliability analysis of digital systems has become an important element of probabilistic safety assessment (PSA). In a reliability analysis of digital systems, fault-tolerant techniques and their effectiveness must be considered. A fault injection experiment was performed on a safety-critical digital I&C system developed for nuclear power plants to evaluate the effectiveness of fault-tolerant techniques implemented in the target system. A software-implemented fault injection in which faults were injected into the memory area was used based on the assumption that all faults in the target system will be reflected in the faults in the memory. To reduce the number of required fault injection experiments, the memory assigned to the target software was analyzed. In addition, to observe the effect of the fault detection coverage of fault-tolerant techniques, a PSA model was developed. The analysis of the experimental result also can be used to identify weak points of fault-tolerant techniques for capability improvement of fault-tolerant techniques.

1. Introduction

Recently, instrumentation and control systems (I&C) in nuclear power plants (NPPs) have undergone digitalization. Deterioration and an inadequate supply of components of analog I&C systems have led to inefficient and costly maintenance. Moreover, since the fast evolution of digital technology has enabled more reliable functions to be designed for NPP safety, the transition from analog to digital has been accelerated. Owing to the distinguishable characteristics of digital I&C systems, a reliability analysis of digital systems has become an important element of probabilistic safety assessment (PSA). Digital I&C systems include software which provides a unique characteristic compared to analog I&C systems. Also, digital I&C systems employ various fault-tolerant techniques to enhance system reliability. The reliability of the software and that of fault-tolerant techniques should be accurately estimated in order to correctly estimate the reliability of digital I&C systems [1–3].

Fault-tolerance is a system's capability through software for helping the system to perform required functions correctly despite the presence of faults. Fault coverage is a measure of the system's ability to perform fault detection, fault isolation, and fault recovery and is mathematically defined as the conditional probability that, given the existence of a fault, the system will detect and recover from the fault [4,5]. If a system failure is detected by fault-tolerant techniques, the failed system does not affect the system operation. For instance, in a reactor protection system (RPS) which has four redundant channels with 2-out-of-4 voting logic, when a channel is failed and detected by fault-tolerant techniques, the voting logic is changed to 2-out-of-3 voting logic. When two channels are failed simultaneously, the RPS generates a trip signal regardless of trip condition according to fail-to-safe design concept. Therefore, in a PSA model, a hardware failure event and the fault detection failure of its fault-tolerant technique are combined with AND gate. A fault-tolerant technique may be an advantageous feature of a digital I&C system. Research has shown that a fault-tolerant technique may improve system safety. A specific fault-tolerant technique,

* Corresponding author.
E-mail address: sjlee420@unist.ac.kr (S.J. Lee).

https://doi.org/10.1016/j.net.2018.11.012

Acronyms

APT	Automatic Periodic Test
ATIP	Automatic Test and Interface Processor
BP	Bistable Processor
CCF	Common Cause Failure
COM	Cabinet Operator Module
CP	Coincidence Processor
CPU	Central Processing Unit
CSD	Component Self-Diagnostics
FV	Fussel-Vesely
I&C	Instrumentation and Control
IDiPS-RPS	Integrated Digital Protection system-Reactor Protection System
KNICS	Korea Nuclear Instrumentation and Control System
MCS	Minimal Cutset
MIAT	Manual Initiated Automatic Test
MT	Manual Test
NPP	Nuclear Power Plant
OS	Operating System
OSD	Online Status Diagnostics
PLC	Programmable logic controller
PSA	Probabilistic Safety Assessment
RAM	Random Access Memory
ROM	Read Only Memory
RPS	Reactor Protection System
WDT	Watchdog Timer

however, cannot detect and recover all possible faults in a system, but it may detect and recover only limited number of faults. Therefore, it is important to quantify the effectiveness of fault-tolerant techniques in estimating the reliability of the system [6,7].

A report published in 1997 by the US National Research Council states that appropriate methods for assessing safety and reliability are key to establishing the acceptability of digital I&C systems in safety-critical plants such as NPPs [8]. A UK Health and Safety Executive guide also points out the importance of the PSA for software-based digital applications as a demonstration of safety [9]. However, there is no widely accepted method for digital I&C PSAs [4]. Conventional PSA techniques cannot adequately evaluate all of the features of digital systems [10]. Kang and Sung found that fault coverage, common-cause failures, and software reliability are the three most critical factors in the safety assessment of digital systems [11]. Kim and Lee identified important factors affecting fault coverage of digital I&C systems based on the literature survey and a fault injection experiment on a digital system running example application software [12].

If the fault detection coverages of a system with multiple fault-tolerant techniques are considered in a PSA model properly, then more accurate reliability of the system could be obtained. This work proposed a method to evaluate the fault detection coverage of fault-tolerant technique using fault injection experiment and provided the result of a fault injection experiment on a safety-critical digital I&C system developed to be applied to NPPs. In addition, to observe the effect of the fault detection coverage of fault-tolerant techniques, a PSA model was developed. An analysis on the result of the fault injection experiment is expected to provide deeper understanding on the effectiveness of fault-tolerant techniques implemented in the target system.

2. Experimental environment

2.1. Coverage of fault-tolerance techniques

Digital I&C systems are designed using various types of fault-tolerance techniques. Although fault-tolerance techniques aim to enhance safety by detecting, isolating, and recovering from faults in a system and eliminate the negative effect of such faults, implementation of more than one fault-tolerance techniques do not always guarantee that more faults will be detected and properly processed. For example, a system with three fault-tolerance techniques, as shown in Fig. 1, has different inspection ranges and coverage. Some faults are detected by only one fault-tolerance technique, and some faults are detected by two or more than two fault-tolerance techniques. Also, some faults are not detected by any fault-tolerance techniques. However, it is possible to enhance the overall fault coverage through an efficient combination of multiple fault-tolerance techniques [2]. To ensure higher reliability of the system, it is important to properly understand the effectiveness of each fault-tolerance technique(see. Fig. 2).

In our work, a fault injection experiment on a safety-critical digital I&C system were performed to have deeper understanding on the effectiveness of fault-tolerance techniques. After examining

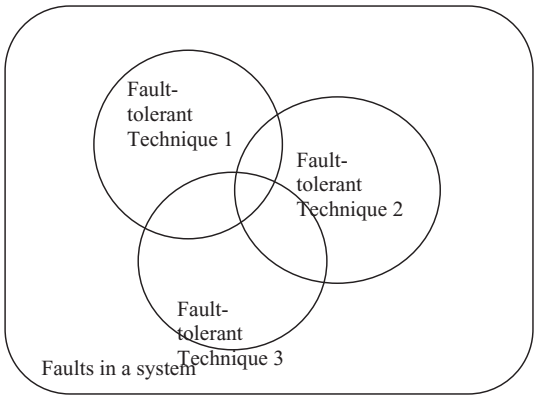

Fig. 1. Coverage of three fault-tolerant techniques in a system [2].

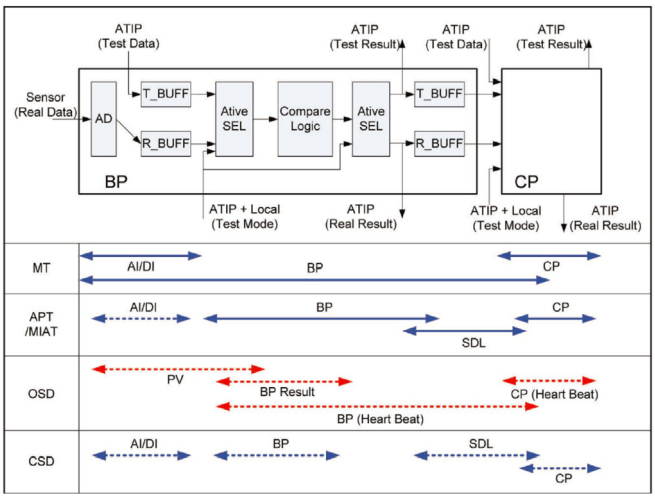

Fig. 2. Fault-tolerant techniques in the IDiPS-RPS [15].

Please cite this article as: M.C. Kim et al., Evaluation of effectiveness of fault-tolerant techniques in a digital instrumentation and control system with a fault injection experiment, Nuclear Engineering and Technology, https://doi.org/10.1016/j.net.2018.11.012

the effect of faults on the output of the target system, faults are classified depending on which fault-tolerance techniques detect the faults.

2.2. Target system

In our fault injection experiment, a prototype of the digital I&C system that was developed to be implemented in a real digitalized NPP were used. The target digital I&C system is the Integrated Digital Protection System -Reactor Protection System (IDiPS-RPS), which was developed in Korea [13,14] under the support of the Korea Nuclear Instrumentation and Control System (KNICS) research and development project. The IDiPS-RPS has four independent channels, where each channel consists of bistable processors (BPs), coincidence processors (CPs), an automatic test and interface processor (ATIP), a cabinet operator module (COM), and other hardware components. BPs receive plant parameters from the sensors and discrete signals from the core protection calculator. BP determines a trip condition by comparing the received plant parameters with the pre-defined trip set-points. The comparison results of BPs are transmitted to four channels of CPs which generate a trip signal based on 2-out-of-4 logic. Based on the outputs of CPs, the trip logic decides a trip condition based on selective 2-out-of-4 logic [15].

IDiPS-RPS has three fault-tolerant techniques as follows [15]:

- CSD: Each programable logic controller (PLC) module has its own self-diagnostic algorithm. CSD represents the diagnostic functions such as watchdog timeout error, execution cycle violation, and instruction operation code error detection functions of a processor module, and loopback monitoring and input/output signal comparison functions of an input/output module.
- OSD: The OSD is performed periodically by the ATIP. OSD includes diagnostic functions such as channel-to-channel comparison of the input signals, setpoint checks to verify the proper setpoint settings, trip status check of BP and CP, and heartbeat check of BP and CP [16].
- APT: The APT monitors the integrity of CP, CP and data link. The monitoring functions of the APT consists of a BP logic test, CP logic test, data link test, and input/output module test

In our work, faults were inserted into a BP of IDiPS-RPS. Among the fault-tolerance techniques of the target system, three were considered: OSD, CSD, and APT.

3. Fault injection experiment

3.1. Fault injection methods

Fault injection is a technique for validating the reliability of a fault-tolerant system. It consists of controlled experiments where the observation of the system's behavior in the presence of faults is explicitly induced. Fault injection techniques can be classified into three main categories [17,18]:

(1) Hardware-implemented fault injection: This is accomplished at the physical level by disturbing the hardware with parameters of environment (heavy ion radiation, electromagnetic interferences, etc.) or by modifying the value of the integrated circuit pins.
(2) Software-implemented fault injection: The objective of this technique is to reproduce at the software level errors that would have been produced upon the occurrence of faults in either hardware or software. It is based on different practical types of injection, including the modification of memory data, the mutation of application software, and the lowest service layers (for example, at the operating system level).
(3) Simulated fault injection: In this technique, the system undergoing testing is simulated in another computer system. Faults are induced, altering the logical values of the model elements during the simulation.

Digital I&C systems contain various types of components such as input/output modules, random access memory (RAM), read only memory (ROM), central processing unit (CPU), and network components. Every component has the potential to cause a fault, and fault-tolerance techniques aim to detect, isolate, and recover from these faults to prevent abnormal behavior in the system. To correctly evaluate the effectiveness of fault-tolerance techniques, the best method is to simulate all possible faults physically by using hardware-implemented fault injections. However, it is difficult to simulate all faults by using hardware-implemented fault injection techniques because this requires expensive hardware, and some faults cannot be controlled and are limited owing to the complexity of the system. In contrast, simulated fault injection techniques require the least cost and time. However, the reliability of the experiment results is low because actual hardware is not used in simulation. While software-implemented fault injection techniques have several limitations compared to hardware-implemented fault injection techniques, it is possible to examine more tests in short time and with less cost. Also, the experiment result reliability is relatively high compared to the simulated fault-injection technique. Therefore, we used the software-implemented fault injection technique in which faults can be injected into the memory area using BP PLC modules. Our fault injection experiment was conducted based on the assumption that all faults in a system will be reflected in the faults in the memory area because a fault should affect the memory area related to the calculation process or variables and cause a wrong output of the system. A fault occurring in any component in a system may have an effect on the calculation process, reading input variables, generating output variables, and so on. A wrong calculation, program halt, variable changes, or wrong execution path may be caused by the fault. Conversely, a fault may have no effect on the output. If a fault does not have any effect on the output, then it is very hard to detect the fault because there are no observable consequences from the fault. If a variable related to the system output is changed by an inappropriate value for the current situation, then the fault may be detectable.

The fault injection experiment was performed based on the following three steps. First, fault types were identified according to the effects of injected faults. Second, a memory area assigned to the BP of the target system was analyzed so that faults can be inserted into only used area and hence unnecessary injection of faults was prevented to reduce the number of fault injections. Finally, the faults were injected into the BP of the target system and the result was analyzed.

3.2. Experimental setup

The fault injection experiment was performed on the memory area of the BP application software. Faults were injected into the memory area assigned to the BP application software by using the Code Composer tool [19]. An automatic fault injection program was developed for the experiment. Fig. 3 shows the environment of the fault injection experiment.

Permanent faults remain until the corrective action is taken, whereas transient faults remain active for a short period of time. Especially, transient faults are a major type of error in computer memory, and in particular, about 98% of RAM errors are transient

Please cite this article as: M.C. Kim et al., Evaluation of effectiveness of fault-tolerant techniques in a digital instrumentation and control system with a fault injection experiment, Nuclear Engineering and Technology, https://doi.org/10.1016/j.net.2018.11.012

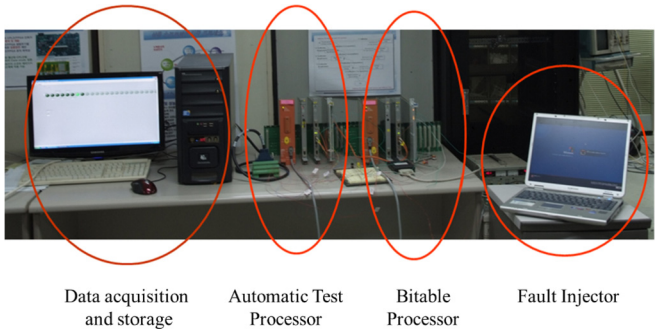

Fig. 3. Environment for the fault injection experiment [15].

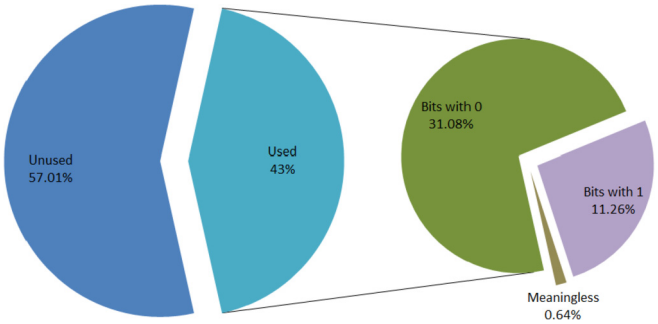

Fig. 4. Analysis on the memory area assigned to BP application software.

errors. Because transient error no longer causes problems after the short period of time, only permanent errors that could cause problems with the RPS are considered. For example, the IDiPS-RPS calculation cycle is 50 ms. Although the RPS fails to generate trip signal at certain time because of a transient fault, a correct output will be generated at the next time step after the transient fault disappears. Since this work is focused on the safety, only permanent faults are considered. For the experiment, only two types of permanent memory faults, stuck-at-0 and stuck-at-1, were injected, because a memory bit has a binary value (0 or 1). The purpose of the fault injection experiment is to examine the effect of faults to the target system and the effectiveness of fault-tolerance techniques. The following limiting conditions were applied in order to reduce the experimental time:

- Only a part of the memory area assigned to the BP was examined. A fault injection experiment with injecting faults into every single bit of the memory assigned to the target system requires a large amount of time and effort because of the large memory size. Each fault injection takes approximately 1 min before providing the result of the fault injection. In fact, a total of approximately 8 million fault injections are necessary just for the memory of the BP operating system (OS) code. Moreover, the memory size of the BP application software is much greater than that of the BP OS. Therefore, fault injections were performed on 3% of used memory area, and only two bits of each 32-bit memory unit (0^{th} bit and 31st bit) were examined. Usually, the first bit (least significant bit) and the last bit (most significant bit) have more significant effect than the other bits because they are used for determining the sign of the data or for checking data integrity such as checksum algorithm. This limiting condition is expected to result in more conservative result.
- The environment for the fault injection experiment is not exactly the same as the actual operating environment. The fault injection conditions differ from plant operating conditions even though actual digital I&C systems were used for the experiment, because the fault injection environment is implemented using only BP and ATIP. If other components were connected, different behaviors could have been observed. With respect to fault-tolerance techniques, it is expected that the result will not be significantly different from those of the actual operating environment.

3.3. Experimental result

Fig. 4 shows the result of the analysis on the memory area assigned to BP application software. From the analysis on the memory area assigned to the BP application software, it was found that 42.99% of the memory assigned to BP application software was used and the remaining 57.01% was not used. It means that inserting faults in the unused area (57.01%) would not have any effect on the function of the BP application software. Only the faults inserted in the used area (42.99%) might have any effect.

The used area (42.99%) of the memory assigned to BP application software can be divided into the bits with 0 (31.08% of the memory area = 72.30% of used memory area), bits with 1 (11.26% of the memory area = 26.20% of used memory area), and meaningless bits (0.64% of the memory area = 1.50% of used memory area). Inserting faults in the meaningless bits will not have any effect on the function of the BP application software. Also, inserting stuck-at-0 faults and inserting stuck-at-1 faults in the bits with 0 and bits with 1, respectively, will not have any effect on the function of the BP application software. Roughly speaking, it is expected that 73.80% (=72.30% + 1.50%) of stuck-at-0 faults and 27.70% (=26.20% + 1.50%) of stuck-at-1 faults will not have any effect on the function of the BP application software.

In the fault injection experiment, 50,980 faults were injected into the used memory area. A half of the faults were stuck-at-0 faults and the other half of the faults were stuck-at-1 faults. Faults were injected into either the 0th bit or the 31st bit of a memory address. Therefore, the injected faults can be categorized into the following four fault groups, each with 12,745 faults:

- Stuck-at-0 faults at 0th bit
- Stuck-at-0 faults at 31st bit
- Stuck-at-1 faults at 0th bit
- Stuck-at-1 faults at 31st bit

Fig. 5 shows the overall result of the fault injection experiment. It turned out that the target system produced correct output (reactor trip) for 90.76% of the injected faults and produced wrong output (no reactor trip) for only 9.24% of the injected faults. Considering that initially 73.80% of stuck-at-0 faults and 27.70% of stuck-at-1 faults, and therefore 50.75% of all injected faults were expected not to cause any adverse effect to the function of the BP application software, 90.76% is much higher than the initial expectation. The high percentage of correct output seems to be largely attributed by the simple binary output of the BP application software. Because the output of the BP application is either 0 (no

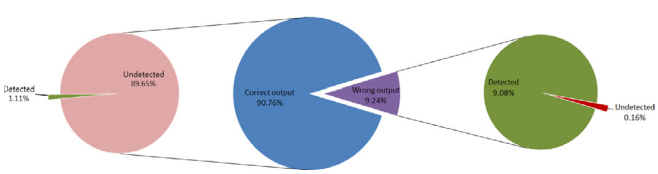

Fig. 5. Analysis on the effect of inserted faults and whether the faults were detected or not.

Please cite this article as: M.C. Kim et al., Evaluation of effectiveness of fault-tolerant techniques in a digital instrumentation and control system with a fault injection experiment, Nuclear Engineering and Technology, https://doi.org/10.1016/j.net.2018.11.012

reactor trip) or 1 (reactor trip), the output is less affected by minor changes in the values on the memory by the injected faults compared to the case when the output is a real number or a combination of real numbers.

Among those injected faults that resulted in the correct output (reactor trip) of the system, 97.77% of the injected faults that resulted in correct output (=89.65% of the injected faults) were not detected by fault-tolerance techniques and only 1.23% of the injected faults that resulted in correct output (=1.11% of the injected faults) were detected. It was already mentioned that about 50% of the injected faults do not change what was originally written on the memory by injecting stuck-at-0 faults on the bits with 0 or injecting stuck-at-1 faults on the bits with 1 or injecting faults on the meaningless bits. Therefore, it can be rather easily explained that those faults were not detected by the fault detection mechanisms. However, the remaining undetected faults (about 40%) actually changed what was originally written in the memory but the final output of the BP application software was not changed, and the injected faults were not detected by any of the three fault-tolerance techniques (OSD, APT, and CSD). Further investigation will be necessary to trace how the change made by those faults did not affect the final output of the BP application software and how those faults could escape the detection by fault-tolerance techniques.

On the other hand, for those injected faults that resulted in the wrong output (no reactor trip) of the target system, opposite observation was made. It was found that 98.28% of the injected faults that resulted in wrong output (=9.08% of the injected faults) were detected and 1.72% of the injected faults that resulted in wrong output (=0.16% of the injected faults) were not detected by any of the three fault-tolerance techniques. While the fault-tolerance techniques were not effective in detecting those inserted faults that resulted in correct output (reactor trip), it was found that they were very effective in detecting those inserted faults that resulted in wrong output (no reactor trip). Further investigations will be necessary on the differences between those inserted faults that changed what was originally written in the memory but resulted in correct output (reactor trip) and undetected by fault-tolerance techniques and those inserted faults that changed what was originally written in the memory and resulted in wrong output (no reactor trip) but detected by fault-tolerance techniques.

From the experimental result, it was found that 99.84% of the injected faults that resulted in the wrong output (no reactor trip) of the target system were properly processed, either by masking or detection, and only 0.16% of the injected faults resulted in the failure of the function of BP application software (generation of reactor trip signal when it is needed), which is possible safety concern. To further improve the safety of the reactor protection system, it will be necessary to analyze in detailed why the 0.16% of the injected faults resulted in wrong output (no reactor trip) without being detected by any of the fault-tolerance techniques.

Fig. 6 shows dependency of the output of BP application software on the fault group (combination of fault type and fault location). It is found that the stuck-at-1 faults resulted in higher percentage of wrong output (no reactor trip) compared to stuck-at-0 faults. It is interesting that the stuck-at-0 faults inserted to 31st bits have very little percentage (0.02%) of causing wrong output (no reactor trip) of the BP application software.

Fig. 7 shows the contribution of each fault group to the number of faults causing wrong output. It is found that a majority of wrong output is caused by the stuck-at-1 faults at 31st bit, while the contribution of stuck-at-0 faults at 31st bit is insignificant. The contributions of stuck-at-0 faults and stuck-at-1 faults are 12.99% and 86.01%, respectively, and therefore the ratio of the two

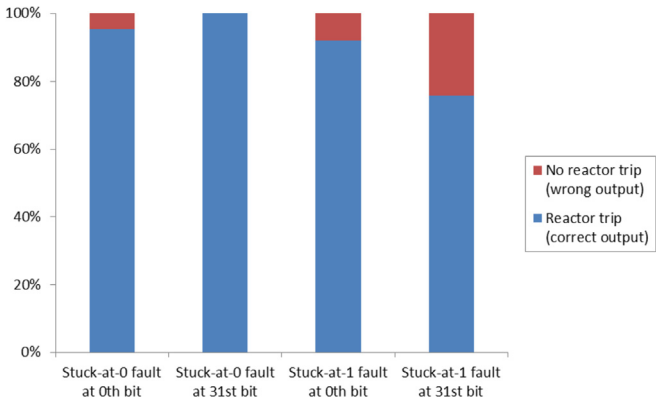

Fig. 6. Dependency of the output of BP application software on the type and location of the faults.

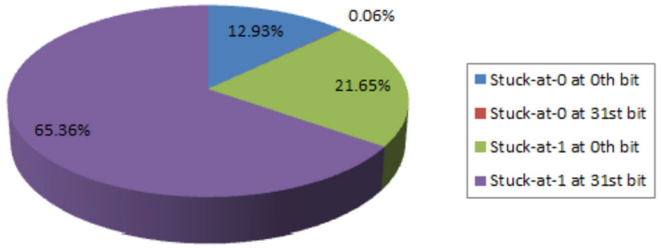

Fig. 7. Contribution of each fault group to the number of faults causing wrong output.

contributions is 4.88. It should be noted that the number of bits with 0 and the number of bits with 1 are 72.30% and 26.20% of the used memory area, respectively, and therefore the ratio of the two is 2.76. By comparing the two ratios (4.88 versus 2.76), it can be shown that possible harmful effect of stuck-at-1 faults might be more significant than that of stuck-at-0 faults. While further analysis should be conducted to identify the reason of the largest contribution of stuck-at-1 fault at 31st bit, it is possible to infer two explanations based on the analysis results. First, as shown in Fig. 4, the portion of bits with 0 is 31.08% and that of bits with 1 is 11.26%. This high portion of bits with 0 causes the larger contribution of stuck-at-1 fault injection. Second, 31st bit is the most significant bit to determine operators or data types. While 0th bit changes the number of data or address, 31st bit changes operators (e.g., AND operator to OR operator) or data types (e.g., integer to float). These may cause the significant contribution of stuck-at-1 fault injection at 31st bit.

To evaluate the effectiveness of the three fault-tolerance techniques, 4711 faults that resulted in the wrong output (9.24% of all injected faults) were classified according to which fault-tolerance techniques detected those faults. It is found that most of the faults (4447 faults = 94.40%) resulting in wrong output were detected by both OSD and APT. 159 faults (3.38%) were detected only by APT and 20 faults (0.42%) were detected only by OSD. CSD detected only 4 faults (0.08%) which were also detected by both OSD and APT. Because of the relatively small number of detections by CSD, further investigation will be necessary on why the effectiveness of CSD in detecting faults is small compared to APT and OSD. As mentioned above, 81 faults (1.72%) that were not detected by any of the three fault-tolerance techniques might raise safety concern and therefore further investigation will be necessary for those faults, too. Fig. 8 summarizes the classification described above.

Please cite this article as: M.C. Kim et al., Evaluation of effectiveness of fault-tolerant techniques in a digital instrumentation and control system with a fault injection experiment, Nuclear Engineering and Technology, https://doi.org/10.1016/j.net.2018.11.012

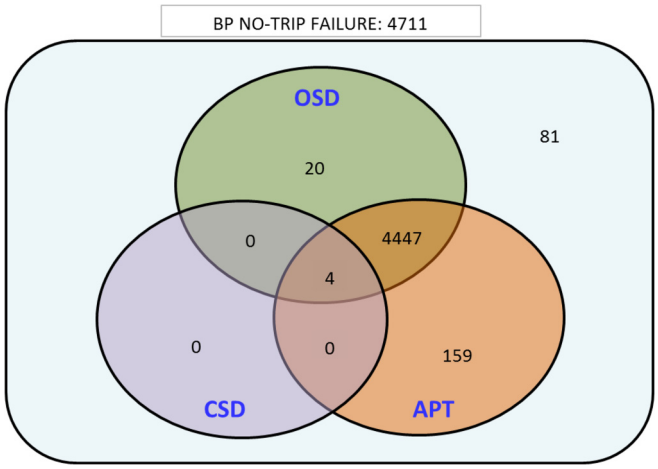

Fig. 8. Classification of 4711 faults that resulted in wrong output of BP application software.

4. PSA model with consideration of fault detection coverage

The effect of the fault detection coverage of a fault-tolerant system on the plant safety can be estimated based on a PSA model. However, most current PSA models consider only fault detection coverage of watchdog timer (WDT) with assumed value (Lee et al., 2015). When the fault detection coverages of all diagnostic functions in a system are considered, more realistic result can be obtained.

Fig. 9 and Fig. 10 show examples of two approaches to reflect fault detection coverages on a PSA model. Both models represent single BP channel failure. If one of the input modules, the processor module, and the output module fails, the BP fails to perform the designated action. The difference between two models is the number of fault detection coverages. While three fault detection coverages are model for three modules in the first model, only one overall fault detection coverage is considered in the second model. Since it is obvious a fault-tolerant technique has its own fault detection coverage, the first approach is appropriate to obtain more accurate result. If the fault detection coverage of each module can be evaluated, the fault tree shown in Fig. 9 will show more valid results. However, it is not easy to evaluate specific fault detection coverage for a fault-tolerant technique because there are intersections among them as shown Fig. 8. As described in the previous study (Lee et al., 2015), duplicated effect reflection on a model should be handled carefully. Moreover, it is expected that the difference between fault detection coverages is relatively low because all fault-tolerant techniques have high fault detection coverage in a safety-critical system. And since only fault detection coverage of the overall system is estimated from the experiment conducted in this work, the second approach is used to observe the effect of fault detection coverage.

Tables 1–3 show the evaluation results with assumed fault detection coverages based on the second approach. Evaluation was conducted on three cases; WDT with 90% fault detection coverage, when all fault-tolerant technologies with 90% coverage, and all technologies with 99% coverage. The tables show the minimal cutsets (MCSs) for the three cases. Since only fault-tolerant techniques considered and their fault detection coverages are different in three cases, the other basic events have the same value. For example, the failure probability of 'Pressurizer pressure measurement loops CCF + Manual reactor trip failure' and 'Trip circuit breakers CCF + Manual reactor trip failure' have the same failure probability in all cases. On the other hand, the failure probabilities related to the failure of digital I&C systems are decreasing as more fault-tolerant techniques considered and their fault detection coverages increase. While the most significant MCS is CCF of RPS output modules in the first case, 'Pressurizer pressure measurement loops CCF + Manual reactor trip failure' is the most significant MCR in the other cases. In the last case, top three importance MCSs are not related to digital I&C systems.

As shown in Fig. 11, the contribution failure probability on demand of RPS for each case is 2.02E-06, 1.03E-06, and 6.58E-07, respectively. Contribution of digital I & C system failure on reactor trip failure is 79%, 40%, and 5%, respectively. It can be said that if a digital I&C system has fault-tolerant techniques with extremely high fault detection coverage, the failure of the digital system can be almost negligible when evaluating the system reliability.

Although the total fault detection coverage of the RPS is not evaluated in the fault injection experiment of this work, the effect of RPS fault-tolerant techniques on reactor trip failure can be roughly estimated based on the evaluated fault detection coverage of the BP. If the RPS fault detection coverage has the similar level of fault detection coverage with that of the BP (98.28%), the reactor trip failure probability on demand will reduce more than 50% of that in a PSA model considering only WDT fault detection coverage. Moreover, the contribution of failure of digital RPS on reactor trip failure will be about 5%. However, to evaluate accurate the overall fault detection coverage of the RPS, large scale fault injection experiment is necessary.

5. Discussion

In this work, the fault detection coverage of digitalized RPS was evaluated based on a fault injection experiment, and its effect on the system failure was estimated using the PSA model. In the result, it was observed that a fault detection coverage has great effect on the system failure probability. Therefore, evaluating fault detection coverages is important for more realistic reliability assessment for digitalized systems. The following issues need to be further investigated to derive more concrete conclusions from the fault injection experiment:

- It is not easy to perform a fault injection experiment for the whole memory area because it requires a large amount of time and effort. In this fault injection experiment, faults were injected into only 3% of the used memory area. This was approximately 1.33% of the whole memory area. Although the fault injection experiment described in this paper provides important insights on the effect of fault occurrence on the function of digital I&C systems and the effectiveness of fault-tolerance techniques, a larger-scale fault injection experiments will be more beneficial in widening our understanding on them.
- The result of the fault injection experiment may be used to quantify the fault coverage of fault-tolerance techniques. The fault injection experiment was performed based on the assumption that all faults of digital I&C components are reflected in the memory, and the failure rate of the memory is not considered. However, different memory bits may have different failure rates because of the loading frequency, addressed variable/code, and so on.
- Different software functions are used according to different scenarios. A function for a given situation can be identified based on the software function analysis. For example, for a trip signal generation, only functions related to the generation of a trip signal need to be considered.

Please cite this article as: M.C. Kim et al., Evaluation of effectiveness of fault-tolerant techniques in a digital instrumentation and control system with a fault injection experiment, Nuclear Engineering and Technology, https://doi.org/10.1016/j.net.2018.11.012

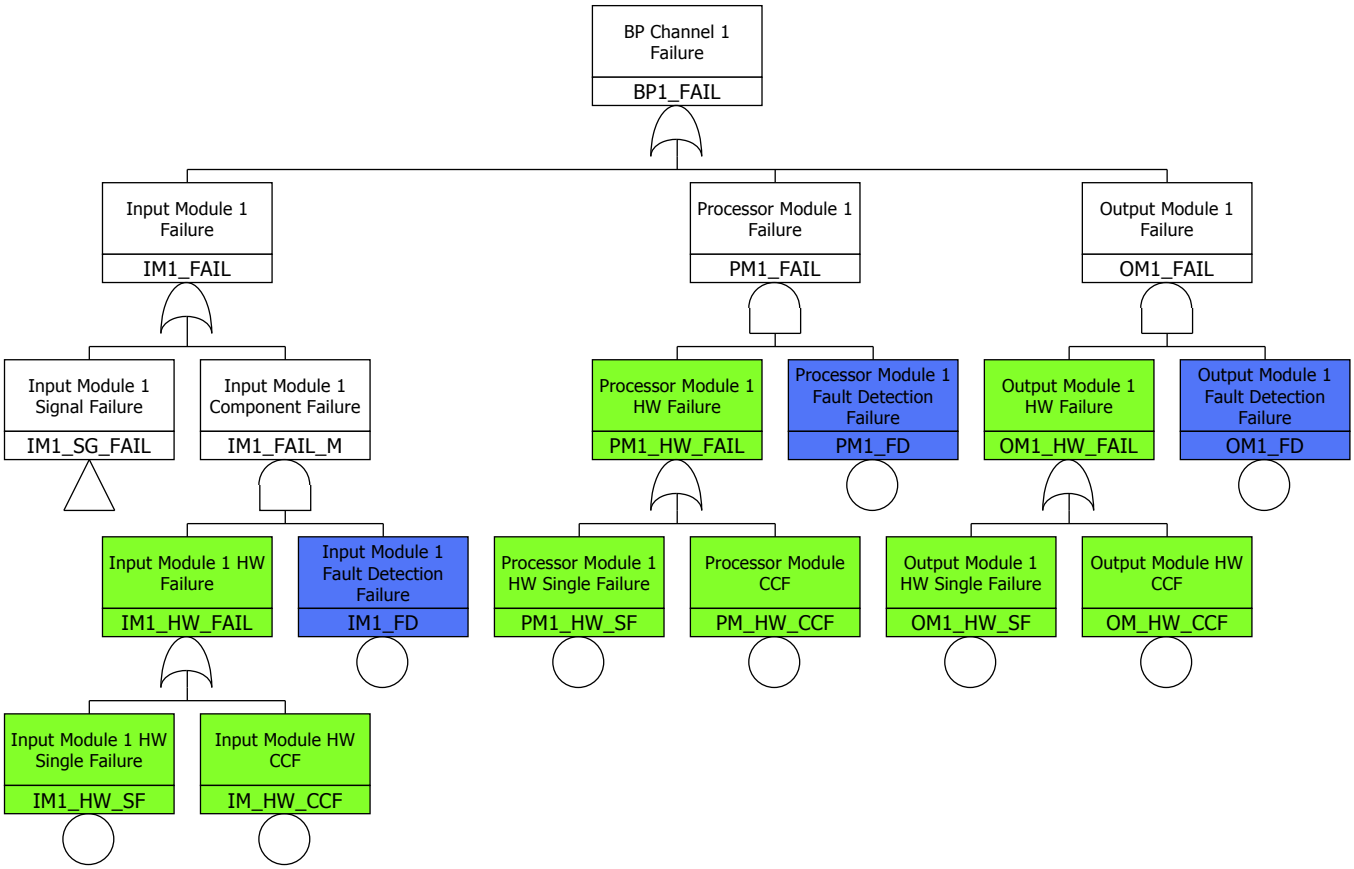

Fig. 9. Fault tree of BP considering specific fault detection coverages.

6. Conclusions

The reliability of software and fault-tolerant techniques should be evaluated to estimate the reliability of digital I&C systems. In the present work, a fault injection experiment on a safety-critical digital I&C system was performed to examine the effect of fault occurrence in digital I&C systems and the effectiveness of fault-tolerant techniques on the fault occurrence. A software-implemented fault injection technique in which faults can be injected into the memory area was used based on the assumption that all faults in a system will be reflected in the faults in the memory area. The fault injection experiment was performed based on the following three steps. First, fault types were identified according to the effects of the injected faults. Second, the memory area assigned to the BP application software of the target system was analyzed for more efficient fault injections. Unnecessary injection of faults was prevented as much as possible to reduce the required number of fault injections. Finally, the fault injection experiment was performed, and the result was analyzed.

From the fault injection experiment, it was first found that the BP application software is very resilient to the fault occurrence in the digital I&C system. More than 90% of injected faults did not cause any adverse effect on the function of the BP application software. It is also important to recognize that some faults actually changed the values written on the memory but did not change the final output of the BP application software.

Second, it was found that fault-tolerance techniques were very effective on the faults that resulted in wrong output of BP application software. Even though only 1.23% of injected faults could be detected by fault-tolerance techniques when the BP application software provided correct output, up to 98.28% of the injected faults were detected by fault-tolerance techniques when the BP application software provided wrong output.

Third, most of the detected faults were detected by both APT and OSD. It was found that 94.40% of the faults resulting in wrong output were detected by both OSD and APT. It seems that some faults cause significant deviation from the normal control flow of BP application software or change the values written in the memory and hence more easily detected by fault-tolerance techniques.

Fourth, possible harmful effect of stuck-at-1 faults might be more significant than that of stuck-at-0 faults. While the used area of the memory consists of 72.30% of bits with 0, 26.20% of bits with 1, and 1.50% of meaningless memory, the contributions of stuck-at-1 faults and stuck-at-0 faults to those faults resulting in wrong output were 86.01% and 12.99%, respectively. It means that relatively higher potential of causing wrong output was observed in stuck-at-1 faults.

Fifth, it is expected that the RPS failure probability decreases much by reflecting the fault detection coverage on a PSA model. From the fault injection experiment, the fault detection coverage of the BP was evaluated as 98.28%. If an RPS has the similar level of fault detection coverage with the evaluated fault detection coverage of the BP, the reactor trip failure probability on demand reduces about 3 times compared to that of the model which only WDT fault detection coverage. Especially, the contribution of digital I&C system failure on the reactor trip failure reduced significantly. Therefore, to obtain more reliable reliability of digital I&C systems fault detection coverage of implemented fault-tolerant techniques

Please cite this article as: M.C. Kim et al., Evaluation of effectiveness of fault-tolerant techniques in a digital instrumentation and control system with a fault injection experiment, Nuclear Engineering and Technology, https://doi.org/10.1016/j.net.2018.11.012

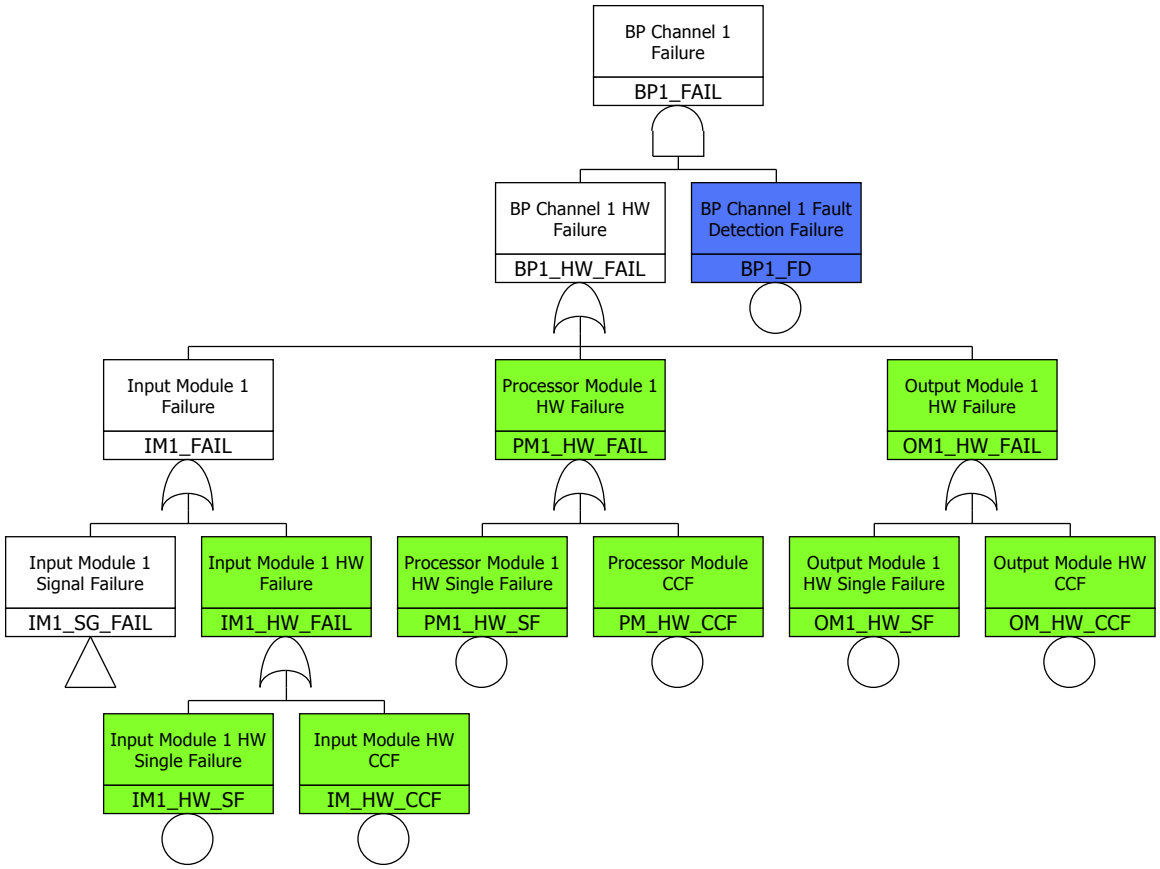

Fig. 10. Fault tree of BP considering overall fault detection coverage.

Table 1
Minimal cutsets for case only with WDT (90%).

Failure probability	Fussell-Vesely (FV) importance	Basic Event 1	Basic Event 2	Basic Event 3	Basic Event 4
7.26E-07	0.360005	CP OM CCF	FD	MB1	
3.70E-07	0.183308	MLKPT	MB1		
3.65E-07	0.181270	BP IM CCF	FD	MB1	
2.06E-07	0.102044	CP PM CCF	FD	MB1	
1.63E-07	0.080960	TCB CCF	MB2		
1.03E-07	0.050958	BP PM CCF	FD	MB1	
6.63E-08	0.032889	PTKPT	MB1		
3.03E-09	0.001505	TYPTA	TYPTC	TYPTD	MB1
3.03E-09	0.001505	TYPTB	TYPTC	TYPTD	MB1
3.03E-09	0.001505	TYPTA	TYPTB	TYPTD	MB1

Table 2
Minimal cutsets for case only with all fault-tolerant techniques (90% coverage).

Failure probability	Fussell-Vesely (FV) importance	Basic Event 1	Basic Event 2	Basic Event 3	Basic Event 4
3.70E-07	0.183308	MLKPT	MB1		
2.06E-07	0.102044	CP PM CCF	FD	MB1	
1.63E-07	0.080960	TCB CCF	MB2		
1.03E-07	0.050958	BP PM CCF	FD	MB1	
7.26E-08	0.070201	CP OM CCF	FD	MB1	
6.63E-08	0.032889	PTKPT	MB1		
3.65E-08	0.035348	BP IM CCF	FD	MB1	
3.03E-09	0.001505	TYPTA	TYPTC	TYPTD	MB1
3.03E-09	0.001505	TYPTB	TYPTC	TYPTD	MB1
3.03E-09	0.001505	TYPTA	TYPTB	TYPTD	MB1

Please cite this article as: M.C. Kim et al., Evaluation of effectiveness of fault-tolerant techniques in a digital instrumentation and control system with a fault injection experiment, Nuclear Engineering and Technology, https://doi.org/10.1016/j.net.2018.11.012

Table 3
Minimal cutsets for case only with all fault-tolerant techniques (99% coverage).

Failure probability	Fussell-Vesely (FV) importance	Basic Event 1	Basic Event 2	Basic Event 3	Basic Event 4
3.70E-07	0.183308	MLKPT	MB1		
1.63E-07	0.080960	TCB CCF	MB2		
6.63E-08	0.032889	PTKPT	MB1		
2.06E-08	0.031264	CP PM CCF	FD	MB1	
1.03E-08	0.015612	BP PM CCF	FD	MB1	
7.26E-09	0.011030	CP OM CCF	FD	MB1	
3.65E-09	0.005554	BP IM CCF	FD	MB1	
3.03E-09	0.001505	TYPTA	TYPTC	TYPTD	MB1
3.03E-09	0.001505	TYPTB	TYPTC	TYPTD	MB1
3.03E-09	0.001505	TYPTA	TYPTB	TYPTD	MB1

* FD: Fault-tolerant techniques fail to detect a HW fault.
* IM: Input module.
* PM: Processor module.
* OM: Output module.
* TCB: Trip circuit breaker.
* MB1: Operator fails to manually generate reactor trip signal.
* MB2: Operator fails to manually trip reactor (case of trip circuit breaker mechanical binding).
* MLKPT: CCF of all measurement loops for Hu PZR pressure.
* PTKPT: CCF of all Hi PZR pressure transmitters.
* TYPTX: Hi PZR pressure transmitter Ch.X fails to provide proper output during operation.

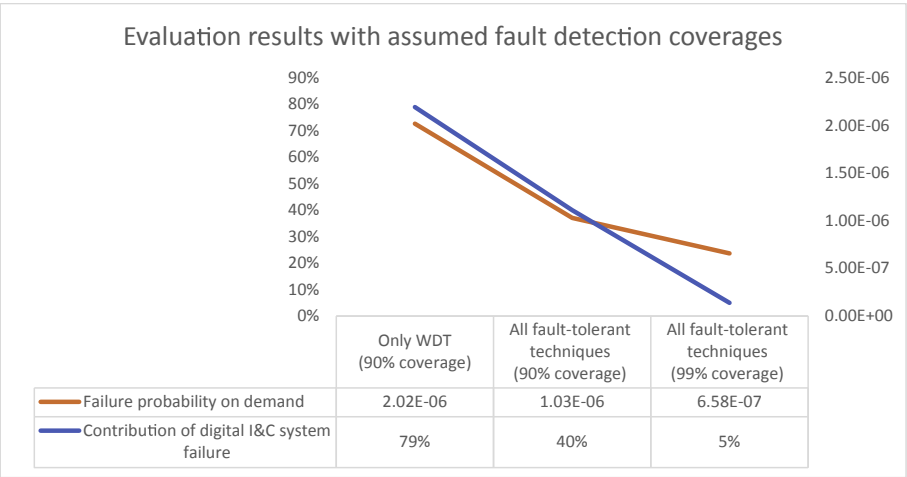

Fig. 11. Evaluation results with assumed fault detection coverage.

should be evaluated and reflected on a PSA model properly.

During the analysis of the experimental result, what need to be further investigated were also identified. It was found to be necessary to trace how the change in memory by those faults resulting in correct output did not affect the final output of the BP application software and how those faults could not be detected by fault-tolerance techniques. It is also important in the safety viewpoint to clearly identify why the 0.16% of the injected faults that resulted in wrong output (no reactor trip) were not detected by any of the three fault-tolerance techniques. Relatively low effectiveness of CSD in fault detection also needs to be further investigated.

Even though there were several limiting conditions in the fault injection experiment and the analysis on it in the perspective of scale of the experiment, depth of analysis, and so on, many important insights could be identified on the effect of faults and the effectiveness of fault-tolerance techniques in a digital I&C system. Weak points of fault-tolerance techniques will be analyzed in more detail so that the results can be reflected in the design to improve the capability of fault-tolerant techniques. A larger-scale fault injection experiment is expected to be very beneficial for this purpose.

Acknowledgements

This research was supported by the Chung-Ang University Research Grants in 2017. This work was supported by the National Research Foundation (NRF) grant funded by the Ministry of Science and ICT (MSIT), Republic of Korea (No. NRF-2018M2B2B1065653). This research was financially supported by Human Resources Program in Energy Technology of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), funded by the Ministry of Trade, Industry and Energy (MOTIE), Republic of Korea. (No. 20174030201430).

References

[1] H.G. Kang, M.C. Kim, S.J. Lee, H.J. Lee, H.S. Eom, J.G. Choi, S.C. Jang, An overview of risk quantification issues of digitalized nuclear power plants using static fault tree, Nucl. Eng. Technol. 41 (2009) 849–858.
[2] S.J. Lee, J.G. Choi, H.G. Kang, S.C. Jang, Reliability assessment method for NPP digital I&C systems considering the effect of automatic periodic tests, Ann. Nucl. Energy 37 (2010) 1527–1533.
[3] S.J. Lee, W.D. Jung, J.E. Yang, PSA Model with consideration of the effect of fault-tolerant techniques in digital I&C systems, Ann. Nucl. Energy 87 (2015) 375–384.
[4] T. Aldemir, et al., Dynamic Reliability Modeling of Digital Instrumentation and

Control Systems for Nuclear Reactor Probabilistic Risk Assessments. NUREG/CR-6942, United States Nuclear Regulatory Commission, Washington, D.C, 2007.
[5] J.B. Dugan, K.S. Trivedi, Coverage modeling for dependability analysis of fault-tolerant systems, IEEE Trans. Comput. 38 (6) (1989) 775–787.
[6] J.S. Lee, M.C. Kim, P.H. Seong, H.G. Kang, S.C. Jang, Evaluation of error detection coverage and fault-tolerance of digital plant protection system in nuclear power plants, Ann. Nucl. Energy 33 (2006) 544–554.
[7] S.J. Kim, P.H. Seong, J.S. Lee, M.C. Kim, H.G. Kang, S.C. Jang, A method for evaluating fault coverage using simulated fault injection for digitalized systems in nuclear power plants, Reliab. Eng. Syst. Saf. 91 (2006) 614–623.
[8] Douglas M. Chapin, et al., Digital Instrumentation and Control Systems in Nuclear Power Plants, National Academy Press, Washington, D.C, 1997.
[9] HSE, The Use of Computers in Safety-critical Applications, HSE Books, London, 1998.
[10] S. Authen, J. Holmberg, Reliability analysis of digital systems in a probabilistic risk analysis for nuclear power plants, Nucl. Eng. Technol. 44 (2012) 471–482.
[11] H.G. Kang, T. Sung, An analysis of safety-critical digital systems for risk-informed design, Reliab. Eng. Syst. Saf. 78 (2002) 307–314.
[12] M.C. Kim, S.J. Lee, Important factors affecting fault detection coverage in probabilistic safety assessment of digital instrumentation and control systems, J. Nucl. Sci. Technol. 51 (6) (2014) 809–817.
[13] K.C. Kwon, M.S. Lee, Technical review on the localized digital instrumentation and control systems, Nucl. Eng. Technol. 41 (2009) 447–454.
[14] J.H. Park, D.Y. Lee, C.H. Kim, Development of KNICS RPS prototype, in: Proceeding of ISOFIC-2005, Nov. 1~4, Tongyeong, Korea, 2005.
[15] J.G. Choi, et al., Fault detection coverage quantification of automatic test functions of digital I&C system in NPPs, Nucl. Eng. Technol. 44 (2012) 421–428.
[16] S. Hur, D.H. Kim, I.K. Hwang, A New Automatic Periodic Test Method for the Digital Reactor Protection System, NPIC&HMIT, Knoxville, Tennessee, USA, 2009.
[17] T. Pinna, L.V. Boccaccini, J.F. Salavyv, Failure mode and effect analysis for the European test blanket modules, Reliab. Eng. Syst. Saf. 83 (2008) 1733–1737.
[18] M. Hsueh, T.K. Tsai, R.K. Iyer, Fault injection techniques and tools, IEEE Comput. 30 (1997) 75–82.
[19] Texas Instruments, Code Composer, User's Guide, 1994.

Please cite this article as: M.C. Kim et al., Evaluation of effectiveness of fault-tolerant techniques in a digital instrumentation and control system with a fault injection experiment, Nuclear Engineering and Technology, https://doi.org/10.1016/j.net.2018.11.012